NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

MICROCOMPUTER NETWORK:
INVESTIGATION AND IMPLEMENTATION

by

Jean L. Egbert

March 1985

Thesis Advisor: Gordon Latta

Approved for public release; distribution is unlimited

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Microcomputer Network: Investigation and Implementation		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis March 1985
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Jean L. Egbert		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93943		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93943		12. REPORT DATE March 1985
		13. NUMBER OF PAGES 77
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution is unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

microcomputer, network, lattice net

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This thesis presents an overview of microcomputer networks in general and the Lattice Net Microcomputer Network in particular. The Lattice Net is a network which has as its ultimate goal the use of power lines as a medium of transmission. The program presented herein is part of an on-going project to implement this system. This program works in conjunction with the code

(Continued)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-LF-014-6601

 SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

ABSTRACT (Continued)

presented in another thesis, "Microlan File Transfer Program for Microprocessors" by Roger D. Jaskot and Harold W. Henry.

Microcomputer Networks: Investigation and Implementation

by

Jean Lorraine Egbert
Lieutenant Commander, United States Navy
B.A., University of California, Davis, CA, 1973

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL
March 1985

ABSTRACT

This thesis presents an overview of microcomputer networks in general and the Lattice Net Microcomputer Network in particular. The Lattice Net is a network which has as its ultimate goal the use of power lines as a medium of transmission. The program presented herein is part of an on-going project to implement this system. This program works in conjunction with the code presented in another thesis, "Microlan File Transfer Program for Microprocessors" by Roger D. Jaskot and Harold W. Henry.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

"No man is an island, entire of itself." (John Donne, 1624). When microcomputers were introduced in 1975, they were small, stand-alone machines which gave people their own "piece of the action". Anyone could be an autonomous computer operator/programmer/analyst/user. People being what they are, however, the first thing they wanted to do was to link to some other machine, either a friend's micro-computer or the big computer at work. From the beginning of microcomputers there has been a need for microcomputer networks, to break their isolation.

For the purposes of this thesis, I will define a network as a system involving:

1. intelligent processors which can stand alone;
2. a medium connecting the processors;
3. some method to control access to the medium.

In addition, the services a network provides are:

1. file transfer;
2. creating a connection to another computer;
3. message transfer, allowing users to converse with one another while both are on the system;
4. entering a job to run on a distant computer.

At the minimum a microcomputer network can be a relatively small program, running on two micros, which transfers files between them. File transfer is one of the most common uses for any network, and is almost impossible without one. This problem is exacerbated when equipment from different vendors is being used. Different types of micros cannot read each other's disks, so electronic transfer is needed. The network makes it easy to acquire a new file, without having the user type it in every time he wants one.

Mainframes, minicomputers, and micros can all be hooked together, if the software creates all the right connections.

Is there much demand for all this communications capability? Almost all offices have a need for computers, and the networks that connect them. Michael Crichton (who recently added software author to his other vocations) says "My sense is that eventually everyone will have a machine for communication. The reason that you will have a computer in your house is the same reason that you have a telephone in your house. It's not an issue; almost no one says, 'why do I have to have a phone?' You're an American so you have a phone in your home - and maybe even in your car." [Ref. 1:p. 28]. There are indications from the marketplace of rapidly rising interest. "...sales of modems for microcomputers have been growing at a phenomenal rate recently; some estimate one million units were sold in the last year, with the curve still headed upward." [Ref. 2:p. 32]. At work there is am impetus to connect the many computers already in place;"...(the) goal is to make all programs, data, and other resources available to anyone on the network without regard to the physical location of the resource and the user." [Ref. 3: p. 3]. In the December 1984 issue of Telecommunication Products and Technology the cover story concerns the latest trend in commercial construction: "intelligent buildings". "In intelligent buildings, all operations are linked through a central, or host, computer. All building systems for environmental control, energy, lighting, fire safety and security are centrally controlled to the extent that they almost run themselves. The intelligent building also contains systems that support telecommunications, data communications and virtually all other information technologies available today." [Ref. 4:p. 1]. In these buildings the wiring and in some cases the microprocessors for networks of various sorts are built in;

tenants need only plug their micros into special wall sockets, the way modular phones work in homes, and they have an instant local network. "In a recent Cross Information Co (CIC) survey, 78 percent of those developers contacted stated plans to build intelligent buildings...As information technologies evolve into a staple item, most commercial developers will provide tenants with such systems and services. Buildings will then need these systems to attract new tenants. A building without such enhancements will be at a serious competitive disadvantage." [Ref. 4:p. 23].

Obviously, microcomputer networks are becoming more than useful and popular. They are becoming an integral part of the way we live and work. As a data communications manager myself, I want to understand what these networks are, how they work, when to use them and why, and what their requirements are, from the inside as well as the outside. There is also a need to understand and cope with a bewildering array of products in the marketplace; there are "over 150 makers of computers, over 100 suppliers of data communications gizmos, hundreds of interconnect companies, 150 local area network vendors, at least 8 carriers for nationwide companies to deal with (out of hundreds offering service), dozens of value-added networks." [Ref. 5:p. W/9]. The number of vendors represents a prediction for the future popularity of networks, as well as a problem for data communications managers: understanding all the offerings. That's why I chose as my thesis the investigation of networks, and the writing of part of a microcomputer network in assembly code. This paper presents what I have learned about microcomputer networks, their uses, and some of the requirements for implementing them, along with the code for part of a microcomputer network, which can run on Northstars or Apple IIs. I have named this program the Lattice Net, in honor of Professor Gordon Latta, Chairman of the Math Department of the Naval Postgraduate School.

II. BACKGROUND

A computer network is a system of hardware and software components that enables the physical and electronic connection of computers. Microcomputer networks perform a small subset of the possible functions, though the sophistication of commercial systems is always growing. They usually perform file transfer, and creating the connection to another computer; sometimes message transfer is performed also. Microcomputer networks are a recent innovation, and they started as relatively simple programs. The physical configuration of a network involves connecting a processor in one computer to a micro via a transmission medium. This can mean a minicomputer with satellite links, or microcomputers connected by telephone lines.

Local Area Networks (LANs) are limited-distance collections of intelligent microcomputers connected by high-speed lines. They involve machines located within an office building or complex, and the farthest distance between two nodes is typically one to ten kilometers. The services offered can be as varied as on any other kind of network. The limitation on distance is caused by attrition of the signal over the cable.

Users and applications at computers, terminals, word processors, and personal workstations must be able to exchange data, send messages via electronic mail, access common databases to massage and manipulate data and generate reports, share programs and applications to speed and reduce development efforts, share expensive storage devices (disks and tapes) and output devices (copiers, high-speed printers, facsimile machines, and

graphics plotters), and conduct these same exchanges with other local and remote networks. [Ref. 9:p. 8]

The Lattice Net project involves the creation of a microcomputer network software package which will run on common microprocessors and will use as a medium of transmission the electric power lines already present in the walls. The Lattice Net is a low cost, low speed, low volume network for initial use at the Naval Postgraduate School. Its functions will include file transfer, data transfer (for example, messages between on-line users), and resource sharing (for example, one printer for several microcomputers). Its range will include any station on the same side of a power transformer, on base or on a ship. The Marine Corps is interested in such a network for use among units operating within about a mile of one another. At the present time the foundation of the network has been written, and is presented in another thesis, "Microlan File Transfer Program for Microprocessors" by Roger D. Jaskot and Harold W. Henry. Their thesis presents the program which performs the physical transfer of data and error-checking. They had to solve some timing problems, caused by the different speeds of different types of microprocessors. The timing differences can be ignored if an interrupt mechanism is used. This mechanism is described in more detail in the description of the program. An interrupt is one of the most sophisticated concepts in computing. This thesis provides that interrupt mechanism, along with the user interface screens. The software is independent of the physical connection between nodes. The nodes can be hardwired together, or connected using phone lines and modems; the ultimate goal is to connect them using the power grid and AC modems.

The Lattice Net will allow multi-directional file transfer. It can act as a modem program, dialing any

computer which has a dial-up port, via a modem. At the present time its speed can be as high as 4800 bps (bps will be explained later). It can currently handle 256 addressees. It will not become a high speed or high volume network.

I will now go over the components of a network, explaining what they are and why they are needed, and then discuss possible systems. I will use the word "node" to refer to a microcomputer.

A. THE COMPONENTS

1. Computers

First off, of course, there are computers. Three components of computers of are special interest to us: the operating system, the communications port, and the internal binary code of the machine.

The operating system is the interface between the hardware and the human user. It presents a "face", with a "personality", to the user. It determines how the system hardware will be used by the applications software. All applications software, including communications software, must be written with a particular operating system in mind, or it won't work. The operating system which has become a de facto standard on 8-bit micros is CP/M (Control Program for Microcomputers, copyrighted by Digital Research Corporation of Pacific Grove, California).

Communications ports are one of the few easy items in our list, because they were standardized several years ago and virtually all microcomputers have the same kind of port. It is the RS232C standard of the Electronic Industries Association (EIA). The plug has 25 pins and the port has corresponding holes. "The EIA RS-232 standard defines the interface between data communication equipment

(the modem) and data terminal equipment (the computer). It does this in four parts, covering voltage, connector size and pin layout, functional signal descriptions, and subsets of the signals for various modems." [Ref. 6:p. 44]. Included in the prescribed standard are such signals as "Request to Send" and "Data Set Ready".

All microcomputers, and all mainframes and minicomputers, except mainframes made by IBM, use ASCII as their internal binary code; IBM uses EBCDIC, which they developed. ASCII stands for the American Standard Code for Information Interchange; EBCDIC stands for Extended Binary Coded Decimal Interchange Code. The two codes are not compatible, and when microcomputers hook up to IBM mainframes the two codes must be translated back and forth. This is usually taken care of by the IBM mainframe.

2. A Medium of Transmission

There are many different ways of getting a signal from one place to another. One is simply two wires twisted together ("twisted pair"). Two are used because wires act as antennae, picking up extra electronic noise, and the signals can be summed at the receiving end; the noise from each cancels the other out. Coaxial cable has shielding to block noise. Optical fiber is coming into more common usage. It can carry a many signals, and the distance between repeaters, which has been about one kilometer, is growing. Microwave transmissions can be used between line-of-sight locations, and satellite links are in common use for trans-oceanic connections.

The Aloha packet radio system uses radio signals as the medium of transmission. It was developed in the early 1970's by the University of Hawaii to link campuses on different islands, and has proved to be both effective and inexpensive. Every site receives every packet sent, and the

receivers filter out those packets not addressed to them. There is one channel in use, which means that senders simply transmit whenever they are ready to, and listen for the returning packet to see if it collided with another one. If it comes back garbled, the sender waits a random time, then transmits again.

One of the decisions to be made concerning media is whether to have a full-duplex or half-duplex line. Full duplex means that both ends of the connection can transmit at the same time without interfering with each other. In half duplex only one is transmitting at any given time. The trick in full duplex is having double connections somehow. On a satellite link, for example, the sender could be transmitting on one channel while simultaneously receiving on a different channel. Two different radio frequencies could be used for one connection in the same manner. The Lattice Net is half-duplex.

Ways of using the media fall into two broad categories, baseband and broadband. "A baseband system is the simplest and most economical type of network. It enables only one device at a time to transmit data. The entire capacity of the system's cable is occupied by each transmission, which can be a limitation if your operation must transmit large amounts of information." [Ref. 7:p. 110]. Ethernet from Xerox is an example of a baseband system. A broadband channel is split into several frequencies, or channels, separated by unused "guard band" frequencies to avoid having signals interfere with one another. Each channel is used for one transmission; several devices can transmit at the same time, using different channels. Cable television and telephone lines are examples of broadband systems.

The ultimate target for the Lattice Net is to make it a broadband system also; the use of AC modems and

frequency regulators on the power grid will allow the use of several frequencies for several channels at once on the net.

Using the power lines for data transfer is not an untried idea. There are no major differences between using power lines and using telephone lines. Some research has already been done, as described in [Ref. 8]. On phone lines, a carrier already exists, and modems impose data on it or remove data from it. With power lines, there is no carrier, so AC modems supply both the carrier and the modulation. The target speed for the Lattice Net is 9600bps. At that speed, 20 Khz per channel will probably be needed, with guard bands of 15 Khz on each side, for a total of 50 Khz per channel. With a range of 100 to 800 Khz, it will theoretically be possible to have 16 channels.

3. Modems

If the telephone system is going to be used for the medium, a method of translating is needed. Computers "talk" in discrete bits, digitally. The telephone system was designed for continuous, analog voices. The computer's digital signal is modulated onto the analog carrier of the phone system, and demodulated at the receiving end, by a modem.

In addition to modulation modems must also handle the speed of transmission. Speed has been referred to as the baud rate, but a more accurate term is bits per second (bps). At slow speeds (110 or 300) they are usually the same, but as speed increases the baud rate and bps rate diverge, and the bps term is becoming more commonly used. Most modems for home use operate at 300 bps, with 1200 becoming more popular as prices have come down. Most people cannot read as fast as 300 bps, so if the user will be reading the screen as a transmission comes in that speed may be adequate. However, banners (the information given when

logging on to a system) and spaces also come across at that speed, and we don't read those at all, so 300 bps is often irritatingly slow. For file transfer 1200 bps is much better. Modems which can be switched to operate at 2400 bps are just now entering the home market.

Another aspect of timing is synchronous vs. asynchronous transmissions. A connection will need one or the other, so modems need to be switchable. In asynchronous transmissions a character is sent down the line preceded by a "start" bit, and followed by one or two "stop" bits. These special bits tell the receiver when the character begins and ends. In synchronous transmissions a timing signal is sent, so the two nodes are synchronized. All the characters are then sent together, with no start or stop bits. Synchronous is faster, but more expensive to implement.

Modems can come with a wide variety of options when they are equipped with their own microprocessors. They can store frequently-called numbers and dial them automatically. They can answer incoming calls automatically. If equipped with a speaker they can let you monitor your calls when dialing, so you can tell if a line is busy or the receiving computer is not answering.

Modems are available in their own boxes, or as plug-in boards which fit into expansion slots in micros.

4. Topology and Access

The topology is the physical pattern of nodes used. There are several different ones; the most common are ring, star, and bus.

In a ring, all the nodes are connected in a loop. Messages are circulated in one direction, and as they pass each node checks to see whether to absorb the data or pass it on. Each node is guaranteed access to the network at

regular intervals, but the failure of a node causes major disruption of the network.

In a star network, there is a central controller; the failure of any node does not affect the network, but if the controller fails it takes the rest of the network with it.

With a bus arrangement control is dispersed to the various nodes, and there is no central controller. Connection to the medium is passive, so the failure of any node does not affect the network. This also makes adding and deleting nodes easy, since it does not cause disruption of the network. The electrical connection, called a "tap", can cause pollution of the medium, however, if it accidently generates white noise on the circuit; this would cause collisions with every message sent. The Lattice Net has a bus topology.

See Figure 2.1 for a graphical representation of these networks.

Access methods are the ways in which nodes get permission to talk. This has to be controlled in some way, or transmissions would collide with one another and become garbled. The simplest method to implement is polling; the central controller asks each node in turn if it has anything to transmit. This is inefficient and slow, especially when only a few nodes want to talk. In token-passing a special string of bits is passed among the nodes; only the node in possession of the token may transmit. This is very efficient in conditions of heavy traffic load.

For situations where traffic is light and "bursty" in nature, as with on-line, interactive terminals, the best method is Carrier Sense Multiple Access with Collision Detection (CSMA/CD). In this scheme, when a node wants to transmit, it listens to the carrier signal. If the line is free, it transmits; if not, it waits, listening until the

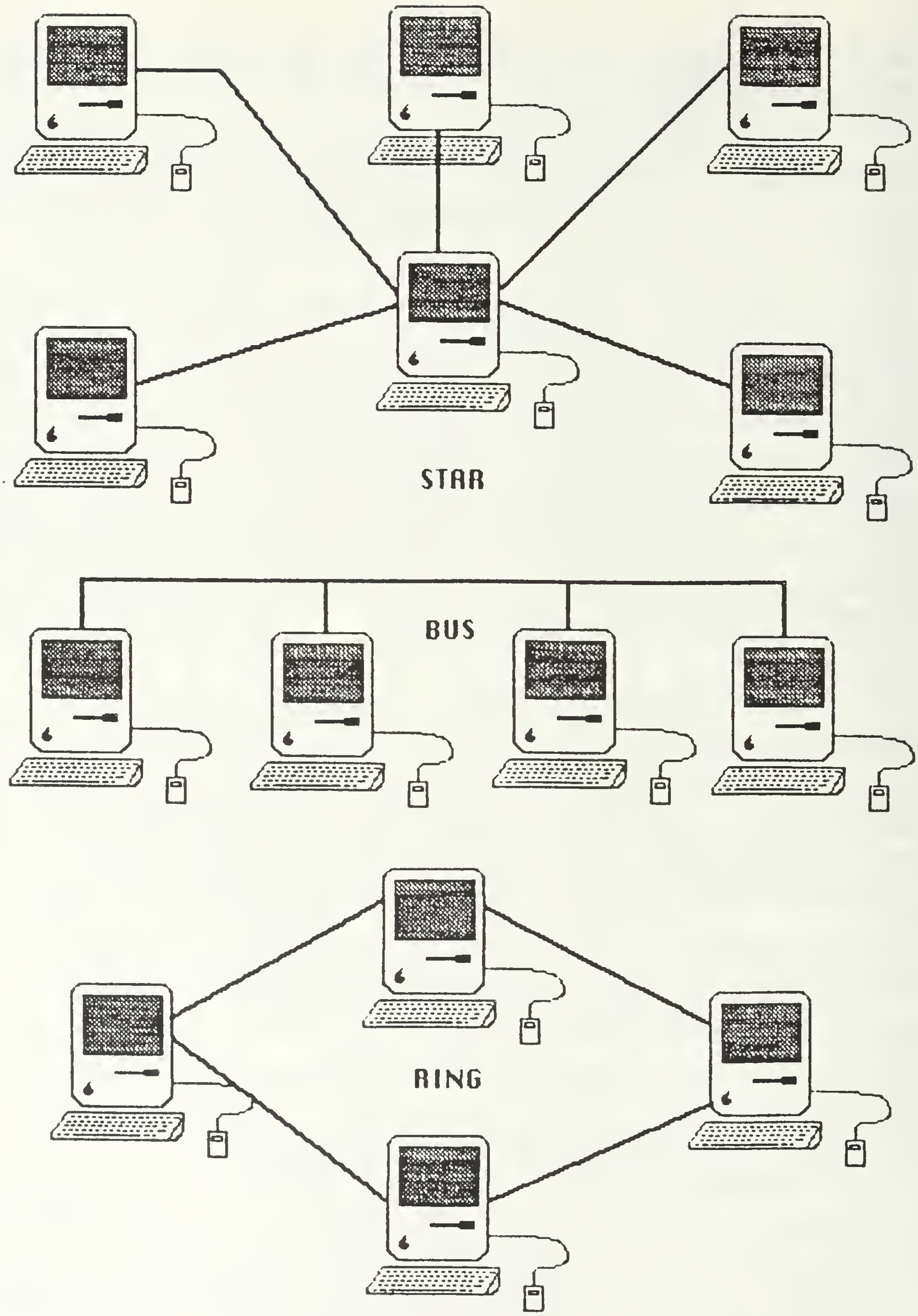


Figure 2.1 Three Topologies

line is free. After transmitting, the node listens for collisions. If it receives its own signal back ungarbled it knows all is well; if the signal is garbled, the node waits a random amount of time, then listens again for a free line, and transmits the same message again. This is the method used on the Ethernet local area network, with good results. It is also the method that will be used by the Lattice Net. The interrupt mechanism can be used to detect if there is anything on the line. If no interrupt signals are received, a node would send, then listen for an acknowledgement. If none is received, the node would wait a random time and send again. Channel One can be designated as being reserved for establishing a connection on one of the other channels. It would be used for initial contact. The node wishing to send would scan the channels to find a currently available one, then inform the receiver, and both would switch to that channel.

5. Communications Software

Communications software is an essential element of any network. It can be relatively simple, or extremely sophisticated, or anything between. It is the brains of the whole operation. As with any software, it is possible to start out simple and add functionality a piece at a time.

There is a basic minimum number of functions which must be performed by a simple network.

1. Bits must be put onto the carrier by the sender, and they must be taken off by the receiver. To have any value these bits must be arranged in an agree-upon order, so the receiver knows which bits are data and which are "housekeeping", and what the data bits mean. Since transmission media are not perfect, noise and errors can be introduced, and the software must make sure that errors are corrected. The

receiver must know when a given transmission is intended for it, and which ones should be ignored. It must know when a message begins, when it ends, and what to do with it. The sender must know what to send, to whom, and when.

2. Some messages are very long, and it is easier to send them in pieces. The software divides them into packets. Each packet must then have a header indicating which piece of which message it is: for example, "2 of 5 of message 386". Receipt of each packet must be acknowledged by the receiver, and lost packets must be re-sent. The receiver then has to guard against duplicates, which can happen if an acknowledgement is lost. See Figure 2.2 for a diagram of a typical packet.
3. Communications software also has to check an incoming transmission for errors acquired during transmission, caused perhaps by electrical noise or power surges. Special mathematical codes, such as Cyclic Redundancy Checks (CRCs), can be used to detect errors. When errors are detected, attempts to correct them can be made, or the packet can simply be re-sent. The Microlan progam of the Lattice Net uses CRCs and re-transmission.

Human beings get involved in all this, too, and the software needs to exchange information with those human beings. A user calls up the communication program, and in some fashion, which varies from program to program, the user identifies himself if he wants to be able to receive transmissions, or tells the software what he wants to send, and to whom.

There are several organizations which have been working to define models for networks, describing the various functions to be performed. One of these is the

SOP	Header	Address	Packet Number	TNP	SOT	Data	EOT	CRC	EOP

SOP - Start of Packet

TNP - Total Number of Packets

SOT - Start of Text

EOT - End of Text

CRC - Cyclic Redundancy Check

EOP - End of Packet

Figure 2.2 A Typical Packet

International Standards Organization (ISO). "In 1978, ISO issued a recommendation to spark greater conformity in the design of communications networks and the control of distributed processing. The recommendation, which has gained wide acceptance, is in the form of a seven-layer model for network architecture, known as the ISO model for Open Systems Interconnection." [Ref. 9:p. 75]. "The network architectures created by most vendors conform to (this model)." [Ref. 9:p. 24].

Why have layers? First, because a network is too big and complex to write as one giant program. Division into layers brings all the benefits of modularity; it is easier to design, write, test, debug, and maintain. Second, the layers which interact with hardware need to include some code peculiar to the hardware interface, such as port addresses, so it is easier to handle if those parts are isolated from the rest of the network software. Third, if it's necessary or desirable to change a function within the network, or perform it in a new way, only the interfaces with other layers need to be standard; what goes on within a layer can be handled in any fashion the designer wishes. Fourth, a network is too big for one person to write in a reasonable amount of time. By dividing it into layers reasonable chunks can be tackled. A network can be functioning and useful with only one layer, and enhancements can be added as they are written.

On a more detailed level there are other protocols to be defined. Will the byte (character) be seven or eight bits long? What will the header look like? What will the parity be? How will errors be detected? These are questions which the data communications manager can leave to the designers of the system.

Linking the computers of different vendors together into a network is a relatively recent development, and many

incompatibilities exist. It would be possible for each vendor to create protocol translators between his machine and those of every other vendor, but this is a grossly inefficient approach. Using a common protocol is a much better idea. There is a profit motive involved in being different, as pointed out above, but if the market demands compatibility the profit motive will work in favor of it. Buyers must insist on getting what is best for them, not what is best for the vendors.

It became apparent, then, that a standard software architecture was needed, and to satisfy that need the ISO designed its Open Systems Interconnection (OSI), a seven-layer model for connecting heterogeneous computers. The functions to be performed in each layer are defined, as well as the interfaces between them. How the functions are performed within each layer is left to individual designers. For compatibility to exist, however, the interfaces must be standard.

The first layer, the Physical Layer, is the only one that interfaces physically with its counterpart on another computer. The other layers interact only with each other on one machine. A diagram of the layers is shown in Figure 2.3, and the layers are described briefly below:

1. The Physical Layer is concerned with transmitting bits over a line. The questions to be answered include voltage levels, how much time one bit occupies, whether the connection is to be half- or full-duplex, and other mechanical and electrical matters.
2. The Data Link Layer handles transmission errors. Streams of data are broken up into frames, and special bits identifying the beginning and end of a frame are added. Acknowledgements are sent and received by this layer, and procedures for avoiding duplicates are also here. Differences in timing must

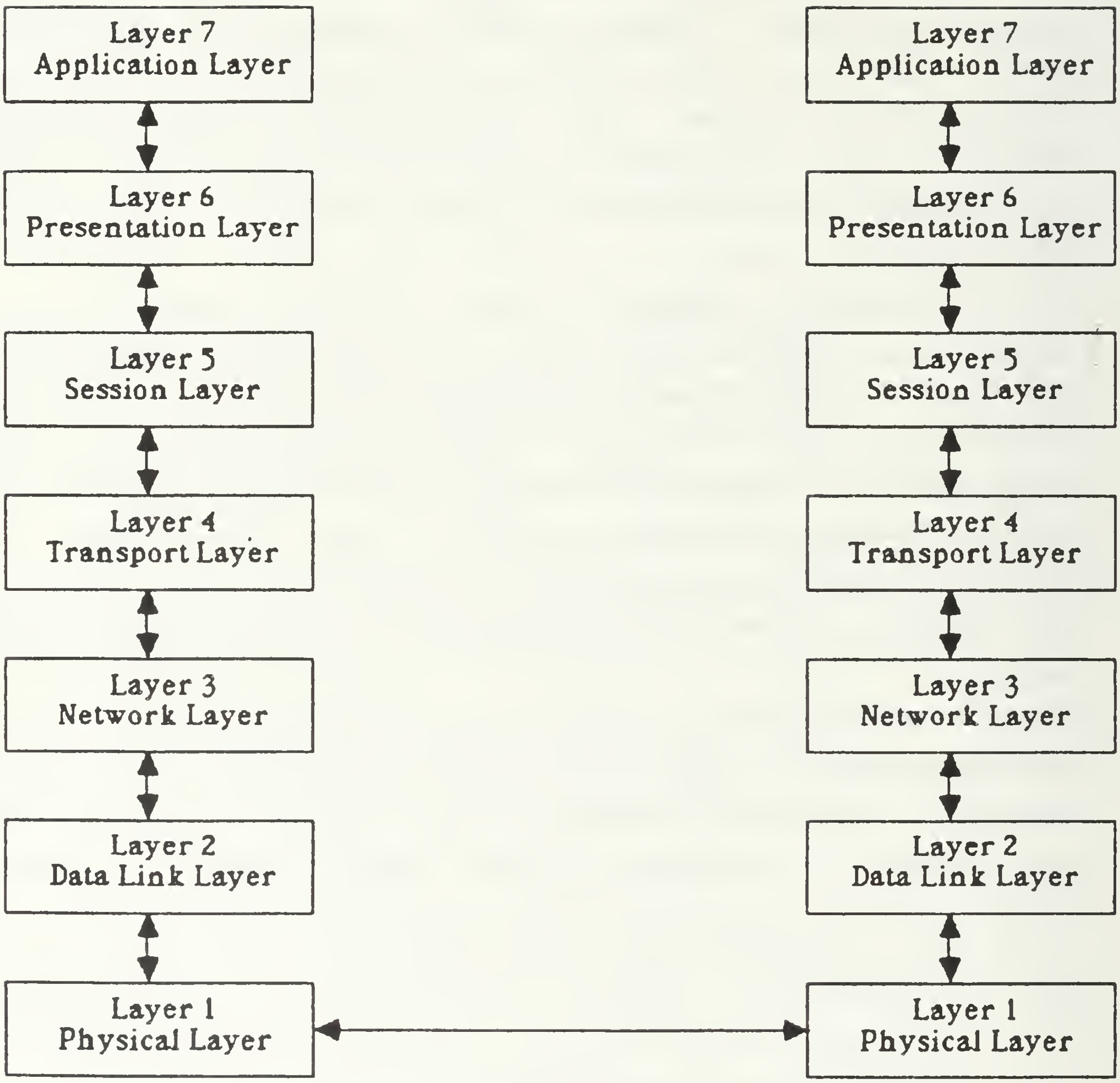


Figure 2.3 The ISO Layers

be dealt with; if one machine is faster than the other, it must be prevented from swamping the slower one.

3. The Network Layer takes care of ensuring that packets are received in the same order they are sent. In sophisticated networks there can be many different paths a packet could take to its destination, so a later packet could arrive earlier than a preceding one. This layer holds such out-of-order packets until earlier ones arrive.
4. The Transport Layer governs end-to-end integrity of transmissions, so that transactions (for example, accesses to a database) do not interfere with one another.
5. The Session Layer establishes a dialog with another machine. If a connection is broken, this layer attempts to re-establish it, transparently to the user. Failing that, this layer closes the connection with as little lost data as possible.
6. The Presentation Layer handles screen displays that interface with the user. Other functions are translation of binary character codes, when needed, text compression, for example, removing (or adding back) blanks or leading zeros, and security.
7. The Application Layer can handle various functions, depending on the user. One example is accessing distributed databases (a database divided up into pieces on more than one machine).

One of the trickiest problems is connecting the equipment of different vendors together. Each will probably use different conventions, and these must be reconciled.

B. THE SYSTEM AS A WHOLE

Now that we have the pieces, how do we put them together? There are many possibilities. The choice must be driven by the needs of the organization.

One of the largest networks is maintained by the Department of Defense. It is the Defense Data Network (DDN). It is a communications network linking universities and military installation throughout the world. Accounts can be established at various organizations, and the user then logs on and works on the computer just as if he were on a local terminal. Messages can be sent to anyone on any computer in the system. The DDN is describes in more detail in the chapter on Military Requirements and Applications.

Local Area Networks (LANs) have been developed in the last few years by several major vendors. Ethernet from Xerox and Wangnet from Wang are two notable examples. They are sophisticated technically, and primarily intended as Office Automation sytems. They began as a way of linking office workstations, so that documents could be created using word processing, then disseminated and filed electronically. The functions usually included graphics such as pie charts and bar graphs, appointment calendars, and small data-base managers. They have not included electronic spreadsheets or a general-purpose computing facility such as a BASIC interpreter or compilers. As they evolved the vendors have added the capability of adding microcomputers to the net; this expands the functionality that was limited before.

Implementing LANs requires much more time and money than do other systems. These are package deals, involving medium, software, operating system, and at least a few terminals, as well as (optionally) printers, as an integrated whole, usually from one vendor. No modems are

needed, since getting the bits onto the wire is part of the package. Even though LANs are usually offered by a single vendor, more and more are advertized as accommodating equipment from a variety of suppliers.

The Lattice Net is a fairly simple network at the present time. It is a micro-to-micro connection, allowing file transfer. With the Lattice Net program, two microcomputers, and a cable with an RS232C plug on each end, a small network can be constructed and files transferred back and forth. The microcomputers can be from several vendors (Apple, Northstar, or IBM), and need not match each other. In the future, the addition of AC modems will allow up to 256 microcomputers to be hooked together, using the medium of the electric wires already present in every modern building. The program is ready for this expansion now.

C. CONCLUSIONS

Commercial software is available to implement networks, with a variety of functions and prices. Relatively simple programs are also widely available in the public domain. Network software is hardware-specific; that is, the program needs to be written for the specific hardware configuration on which it is to be used.

The choice of a system must be based on the needs of the organization, both present and future. Since it is often difficult to anticipate needs, the best approach is to develop a flexible system that can grow as requirements do, and to recognize that the technology is in a very young and dynamic state. Policies and procedures are still being formulated, and will continue to change as technology evolves. The industry is in an experimental stage, and it is best to acknowledge that and participate in it. Because of the wide variety of systems, networking can be tried on

several levels of sophistication, and several systems can co-exist in the same organization. It may be that the best "final" solution (as final as automation decisions ever are) will include more than one system, to suit several purposes.

III. STANDARDS

A. INTRODUCTION

"God knows this market is a mess." [Ref. 7:p. 104] In the January 1985 issue of Government Computer News (a bi-monthly newspaper) there is an advertisement for the 7th Annual Communication Networks '85 Conference and Exposition. It includes 750 exhibit booths and over 200 companies. The list of participants includes the giants of the computer, telecommunications, and electronics industries: AT&T, COMSAT, Digital Equipment Corporation, GTE, Hayes, IBM, and RCA; there are also major organizations in the field, such as the IEEE and The Yankee Group (a research consulting firm). Such a collection of equipment and techniques is mind-boggling, and the potential for confusion is enormous. By "confusion" I mean both confusion of the part of users and the Biblical "confusion of tongues". Without standard protocols and interfaces there would be little telecommunicating going on among the equipment of so many vendors. Standards are as essential to this field as is a lingua franca to the world of diplomacy. Without them we would have the Tower of Babel revisited.

In some respects, this is exactly what has happened in the computer field. Because of the way it developed historically, computers could not talk to one another most of the time. The development of computers can be contrasted with the development of the telephone. From the early days of the telephone there was a need for interconnection. When there were many companies in the business different systems were incompatible. The problem was solved by giving AT&T a

monopoly. When competition was gradually allowed again, all other companies conformed to AT&T standards. Everything from voltage levels to phone numbers follows their practices. In the computer industry, on the other hand, computers originally were stand-alone devices. Different companies evolved different ways of doing things, and the equipment of one manufacturer often could not talk with that of another.

> Further, suppliers who enjoyed large market shares soon found it advantageous to maintain their own special or proprietary communications standards. The use of proprietary standards therefore locked the customer into certain equipment. Today there is a wide variety of communications techniques among computers and between terminals and computers because of these proprietary communications standards. [Ref. 11:p. 6].

This is not to say that no standards exist. Computers did not spring full-grown from the head of Zeus. They emerged from the electronic industry, which had already discovered the need for standard interfaces such as electric outlets and voltage levels. As a result, several hardware standards have been created and widely adopted, and software standards are slowly arising also.

What is a standard? In the computer industry we speak of two kinds of standards: the kind that are recommended or established by committees, and de facto standards. Often the former are simply official versions of the latter. At other times the committees attempt to bring order out of chaos by leading the way in establishing a standard way of doing something. De facto standards arise when the buying public adopts something in large quantities, for any of a variety of reasons. Perhaps the company is large and influential. Perhaps the product was simply first, and a large

base of users became established before any competing product came along. Market pressures are sometimes brought to bear. For example, the CP/M operating system was the first one to be written for a microprocessor but not a particular vendor. Any vendor with an 8080 microprocessor could simply buy a license to sell CP/M with a computer, and a base of software already existed for it. As more CP/M systems were sold, more software was written for it, encouraging more vendors to use it, etc. The two sides, hardware and software, fed each other, and a de facto standard was born.

There are several standards organizations, both American and international. They are described in Table 1.

1. Hardware Standards

One of the hardware standards used by virtually every micro is the communications interface, EIA's RS232C, discussed previously. Another, which arose from the marketplace, is the Bell 212A specification for modems operating via an RS232 port. This standard is not compatible with the corresponding V.22 specification created by CCITT, nor was it first in the market. However, "...(its) impact was similar to that of the IBM PC entry on the personal computer market. Because of Bell's size, reputation, and influence, its standard was quickly established." [Ref. 6:p. 45] Nor is it likely that a conversion to the CCITT standard will occur. "Regardless of standard changes, Hank Morgan, product line manager for Gandalf Data Inc, a Wheeling, Illinois, modem manufacturer, feels that '...I suspect we're not going to see 212A modems being exchanged for CCITT V.22 modems at all. The switch to CCITT compatible standards in the U.S. will occur as speeds increase and new models with more capabilities appear.'" [Ref. 6:p. 45]

TABLE 1

Standards Organizations

CCITT - The Comite Consultatif International Telegraphique et Telephonique is a committee of the United Nations agency, The International Telecommunications Union. Two study groups within the CCITT develop data communications standards. The standards produced by the CCITT study groups are international versions of the standards produced by the EIA. CCITT has representatives from 84 countries as well as large companies in the electronics and communications industries; these include Western Union International Inc., AT&T, RCA Global Communications Corp., and Nippon Telephone and Telegraph Public Co.

EIA - The Electronics Industries Association is an organization that represents American manufacturers. The EIA publishes standards such as RS232C and RS449 (a 37-pin connector primary and a 15-pin secondary connector) that govern the electrical characteristics of connections between the personal computer and external peripherals such as printers and modems.

IEEE - The Institute of Electrical and Electronic Engineers is an American professional group that establishes electrical standards. The organization has a microprocessor standards committee that sets electrical and electronic standards for the design of microcomputer components and systems.

ISO - The International Standards Organization is a worldwide group composed of standards organization representatives from member nations. The American National Standards Institute (ANSI) represents the United States. The ISO develops international standards for data communications. A seven layer model was developed by this organization to define a universal architecture for interconnecting heterogeneous computer systems.

[Ref. 10:p. 14], [Ref. 6:p. 44]

2. Military Standards

Military networks have several requirements in addition to those for civilian networks.

> These military requirements generally have not been accommodated in standards developed for the civil community and, for the most part, are not even considered in that arena. The use of such standards by the DoD, therefore, would reduce the performance of military data networks to some degree. Reasonable compromises may, however, be possible. [Ref. 12:p. 320].

These requirements include:

1. Survivability - military networks are subject to attacks of many kinds. In addition to the more spectacular threats such as hand grenades and bombs, there are also the subtle ones such as sabotage and electronic jamming. Redundant systems, multiple paths, and Electronic Counter-Counter Measures (ECCM) are some methods for ensuring survivability of the system.
2. Security - our systems must provide protection against electronic eavesdropping.
3. Precedence and Pre-emption - our systems must be able to accommodate emergency traffic by assigning precedence to each message, so that higher-priority traffic can pre-empt lower priority traffic. Such a system is already in existence for Autodin and Autovon.
4. Our networks must interface efficiently with a wide variety of other nets, both tactical and non-tactical.
5. They must be capable of operating in a broadcast mode, to reach dispersed units simultaneously.
6. They must be capable of easy expansion and upgrade. [Ref. 12:p. 320].

The Department of Defense has its own agencies working on the development of standards for military networks, and works closely with civilian agencies also.

A very important reason for the DoD use of commercial standards is that the DoD relies on commercial facilities to a very great extent both during peace time when a great many of these facilities are leased, and during periods of enemy attack when commercial facilities play an important role in restoration and reconstitution of military networks. The use of commercial standards obviously eases these situations considerably for the

DoD. For these reasons, the DoD has adopted the policy of utilizing commercial standards to the greatest extent possible except where the use of such standards will compromise critical military requirements. [Ref. 12:pp. 320-321].

3. Conclusions

When you want to make a phone call to Paris, you pick up the phone, dial a number, wait a few seconds, listen to a few clicks, then the phone on the other end rings. It doesn't matter that there may be several transmission media involved (landline to New York, perhaps, then satellite over the ocean), and the phone systems of two different countries. The connection is made, and stays made until the user is done, with little effort by the caller.

This is the way it should be if you want to call a Parisian computer, too. At the present time it is not, but the technology is still evolving. However, the consistency of standards existing in the phone system may not be achieveable for computers, or even necessary or desireable. Gateways can be used to connect networks which use different protocols. A gateway is a point of connection between two different systems. It is a processor running software which translates differing protocols between the systems. Also, there are different applications for networks, and small ones may have no need to interface with larger ones. The Lattice Net is just such a network, so it does not need to conform too strictly to the ISO model.

None of the major network architectures now in existence, such as Arpanet, SNA from IBM, or Decnet from Digital Equipment Corp., conforms exactly to the ISO model, though there are several points of correspondence, and most of the functions defined in OSI are performed. Conformity was the goal of the ISO, but some authors feel this is too

restrictive; "...total interchangeability of layer N protocols is unnecessary." [Ref. 13:p. 309]. Different local arrangements of computers and LANs are already in place, and different vendors design their systems for special purposes. This flexibility is desireable, and connections can still be made using gateways where needed. "No single technology is ideal for all applications, yet the full collection of systems must interoperate." [Ref. 13:p. 309].

Data communications managers need to remain flexible. "Expect multiple standards." [Ref. 5:p. W/15].

> All may decry the chaos of proliferating devices that are incompatible. However, the day-to-day business and operating procedures for most employees are unfavorable to doing anything about it. On the contrary, existing budgetary procedures that foster departmental self-determination--which are good for general business management-- fly directly at the face of the need for corporate-wide telecomm standardization. As a result, most industry analysts have given up predicting the eventual predominance of one local networking technique over another. They will all be used.... [Ref. 5:p. W/17].

By remaining open to new ideas, while at the same time demanding as much standardization as is practical and feasible, we can develop the best systems for our particular needs.

IV. LATTICE NET PROGRAM DESCRIPTION

This chapter will describe the Lattice Net program, first as the user sees it, then from the programmer's point of view. A copy of the code is included as Appendix B, and a diagram of the program is included as Appendix C.

The Lattice Net is not a stand-alone program; It works with the Microlan program presented in the Jaskot-Henry thesis. That program provides the physical transfer of bits, and error-checking. Both programs will be combined into one as the project progresses, and it must be loaded into every node on the network. Figure 4.1 shows this relationship.

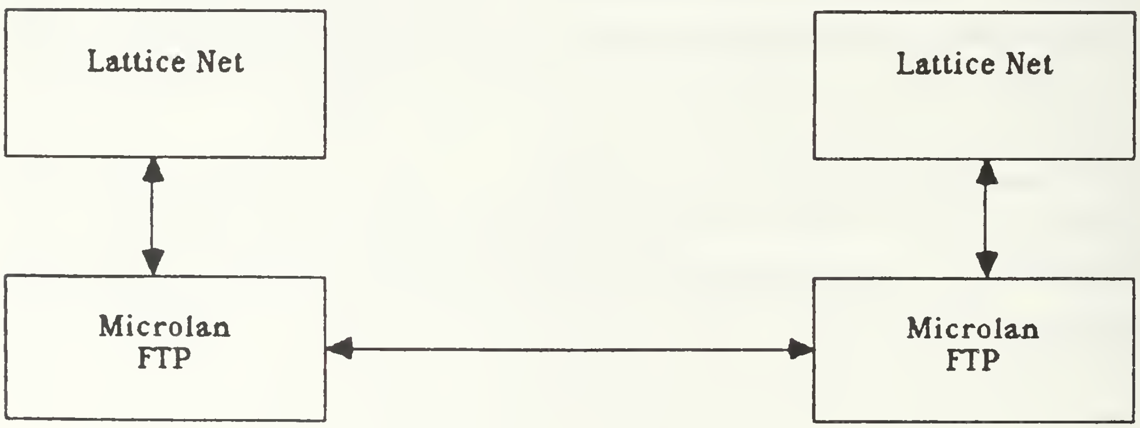


Figure 4.1 The Two Programs

The program written for this thesis presents an interface to the user, and uses the interrupt mechanism to handle incoming files. The code was written using top-down structured design. Individual functions, such as clearing the screen or printing a line, are separated into modules.

A. FROM THE USER PERSPECTIVE

To use the network, it is necessary to turn on the microcomputer, then insert a diskette containing the operating system and a copy of the Lattice Net program; the program will be in a file with a filename of "NET". After booting up the operating system, the user types "NET" to invoke the network. The program responds with a banner and a small menu. The user can choose to Send a File, Enter Receive Mode, or Exit the program.

1. If the user elects to Send a File, the program asks for the one-character address of the terminal to which he wants to send a file. One character allows for up to 256 terminals to be listening on the net at the same time; it could easily be increased if desired. The addresses must be agreed upon among the users ahead of time. When two microcomputers are connected directly addresses are, of course, superfluous; but the program is written with expansion in mind. The next question asked is the name of the file to be sent. The file is transferred, and the menu is redisplayed.
2. If the user elects to Enter Receive Mode, the user is asked for the one-character address of his terminal. Interrupts are enabled, and control is returned to the menu. The user may now exit the program, but the interrupt handling routine is still stored in high memory (location 2000H). As long as it is not over-

written by another program, the user can do anything else he chooses, and still be able to receive files whenever they are sent. The interrupt handling routine can be relocated still higher if that is needed. When a file comes in, the current status of the CPU is saved and control is passed to the interrupt handler.

3. When a user has an incoming file, a small block of the screen is cleared in the upper right-hand corner, and a message to the user is written, informing him that a file is coming in and asking if he wants to receive it. If he says no, the subroutine quits. If he answers yes, control passes to Microlan for the transfer, then back to the interrupt handler. The corner of the screen is cleared, and the interrupted program is restored.
4. If the user selects the "EXIT" option on the menu, the operating system is rebooted.

Figures 4.2, 4.3,and 4.4 show the dialog with the screen.

B. FROM THE PROGRAMMER'S PERSPECTIVE

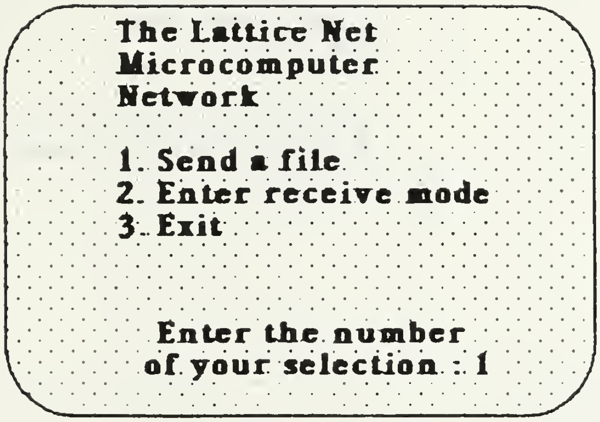
1. Constants

The program starts by defining a set of constants. These are used to make the program more readable; for example, "STAR" is easier to understand than the ASCII code "2AH".

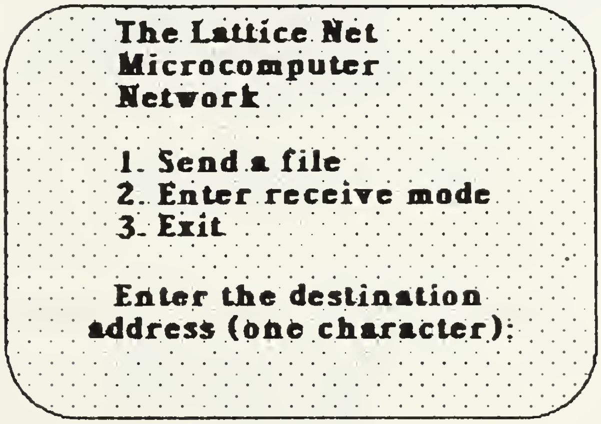
2. MAIN

The MAIN subroutine has three commands, which offer an overview of the whole program.

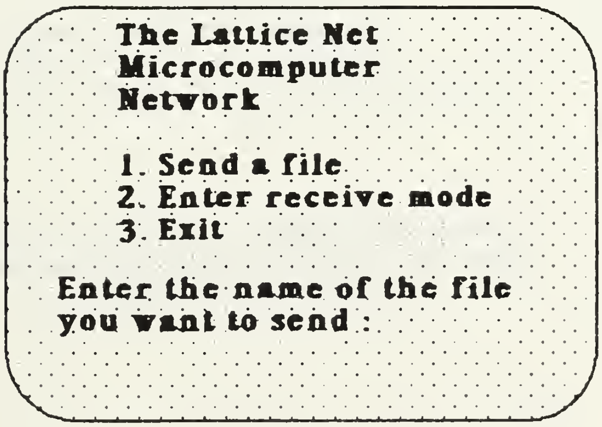
1. INIT initializes the File Control Block (FCB). This is the place in main memory where the filename is



First Question

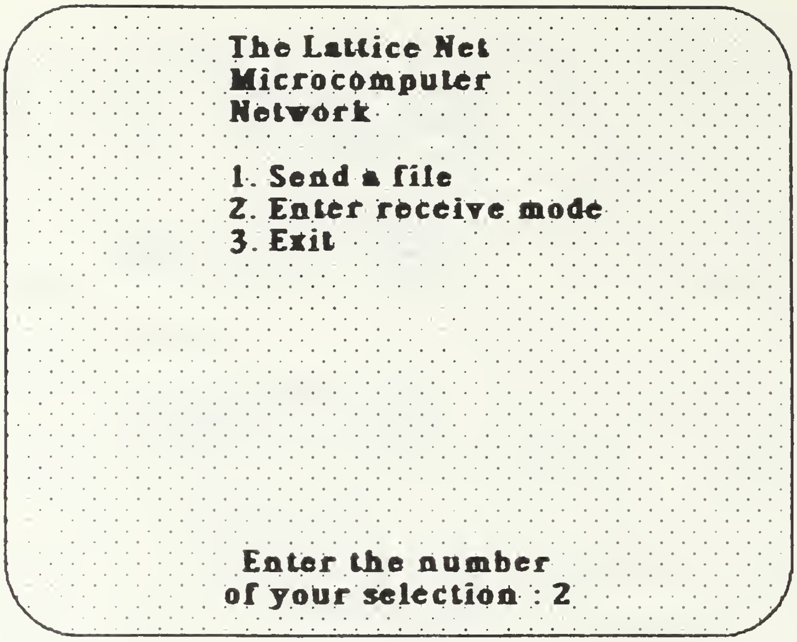


Second Question

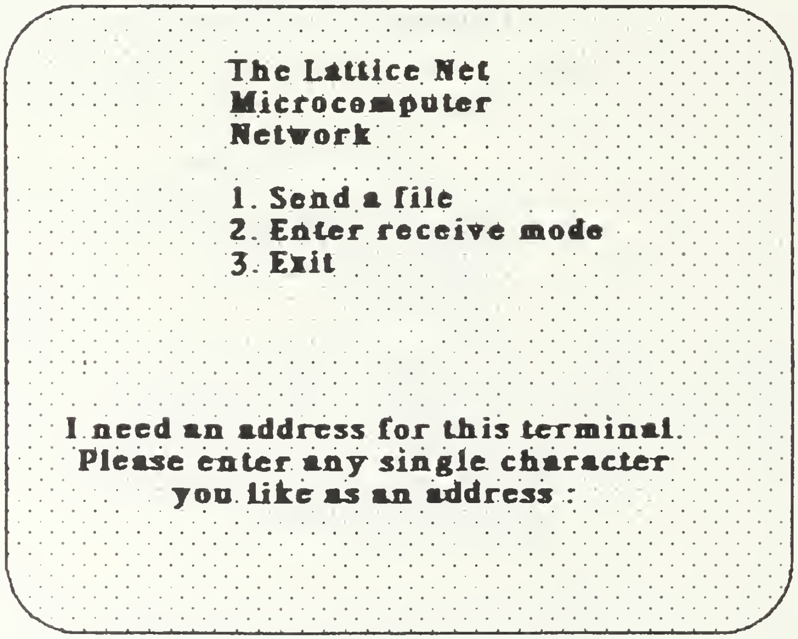


Third Question

Figure 4.2 Sending a File

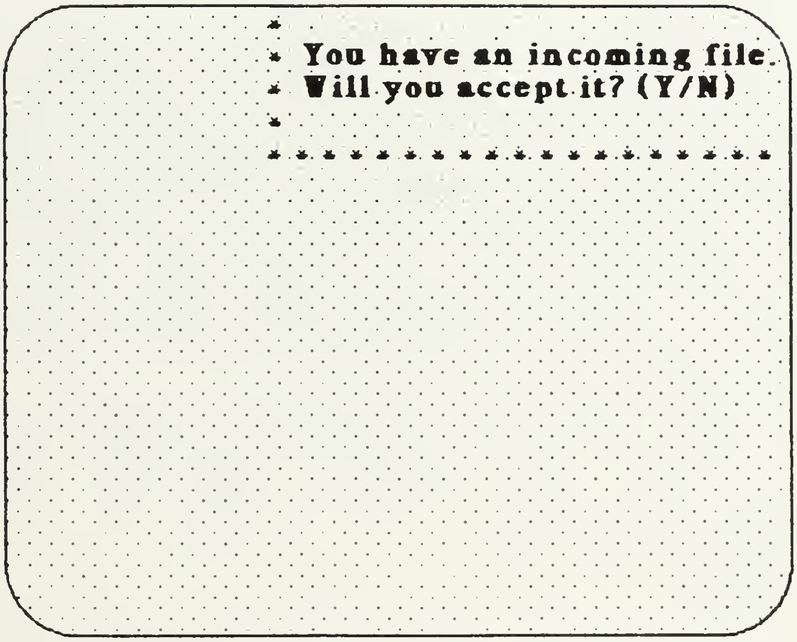


First Question



Second Question

Figure 4.3 Entering Receive Mode



Upon Receipt of an Incoming File.

Figure 4.4 In Receive Mode

stored. The FCB is filled with blanks by this routine.

2. MENU displays the menu on the screen, determines the user's choice, and calls the appropriate subroutine to handle that choice.
3. The third command,"Exit", causes control to pass to the beginning of main memory, which causes the operating system to re-boot.

3. The Menu

The MENU subroutine calls the CLRSCRN subroutine, which clears the screen. Then it prints the menu using the PRSTR ("print string") subroutine, and retrieves the user's answer with the subroutine GETCHAR ("get a character"). The answer is compared with the characters "1", "2", and "3".

1. If it is a "1", the subroutine SENDF ("send a file") is called.
2. If it is a "2", the subroutine RECEIVE is called.
3. If it is a "3", control returns to MAIN and the program is ended.

4. SENDF

The subroutine SENDF requests the address of the destination node, and stores the answer in memory. One byte allows 256 possible characters, for 256 potential nodes. The FCB is re-initialized, in the event this menu selection was made before and a previous filename is still there. The name of the file to send is then requested, and it is stored in the FCB. At this point control is passed to the Microlan program for the actual transfer. At the end of the transfer control returns to MENU.

5. RECEIVE

In this subroutine the Interrupt Mode is set to 1. Using this mode causes an automatic branch to address 0038H, where a jump instruction is stored followed by the address of the interrupt handler. Then the address of the trminal is requested and stored, interrupts are enabled, and control returns to MENU. When an interrupt is generated by an incoming packet, the interrupt handler is invoked.

6. The Interrupt Handler

1. The interrupt handler is located at 2000H; this was a fairly arbitrary choice, and it can be placed higher if desired by changing the ORG statement. Interrupts are disabled, and the registers are saved for returning to the interrupted program.
2. The incoming bit stream is then examined. The first two bytes are checked to see if they are the packet header, which has been defined as 0101. Since Receive Mode could have been entered while a bit stream was already on the line, an interrupt can be generated in the middle of a transmission. If the bits are not a header, therefore, the registers are restored and interrupts are enabled again. Control returns to the interrupted program.
3. The next byte is the address. It is compared with the one that was stored. If they do not match, the interrupt handler is exited.
4. If the addresses match, a message announcing an incoming file is printed in the upper right-hand corner of the screen. The user is given the choice of accepting it or not. If the user answers "N" for No, the corner message is erased, the incoming file is discarded, and the interrupt handler is exited.

If the answer is "Y" for Yes, control passes to the Microlan program for the transfer. When the transfer is complete control returns to the interrupted program.

7. Addressing the Screen.

There are several subroutines involved in addressing the screen, either for sending displays or retrieving answers to questions. Whenever possible, resident operating system subroutines were used for these functions, since they already exist and there was no need to re-write them.

To send a character to the Televideo terminal, it is necessary to address the cursor. This is done by sending an Escape sequence, that is, the Escape code (in ASCII, 1BH), and an "=" (ASCII 3DH). Then you send the row number, then the column number. The upper left-hand position of the screen (often called "home") is 20H, 20H; that is, an offset of 20H is added to the row and column positions.

On the Apple CRT, the situation is similar, with a small variation in the codes. The Escape sequence for addressing the cursor is Escape followed by an ASCII 1EH. The offset is 32H, and the cursor position is sent column first, then row, contrary to the normal order.

There was one exception to the pattern of using resident subroutines. The resident subroutine for writing a string to the screen uses the "$" as a sentinel to indicate the end of the string. When writing the message in the upper right-hand corner of the screen, row 4 is addressed; with the offset, it becomes 24H. In ASCII, 24H is the code for "$", which means that if the resident subroutine is used the display is cut off when that row is reached. Therefore, a subroutine was written essentially duplicating the resident subroutine, but using a " as the end-of-string sentinel. That routine is included as part of the interrupt handler.

With the exception of the differences in addressing the cursor, the code is the same for the Northstar and Apple computers. It is designed to be as portable as possible to other microcomputers.

C. SOME OF THE PROBLEMS ENCOUNTERED

1. Screen creation in assembly code requires knowledge of cursor addressing. Discovering the offset for the home position, and whether row or column is designated first, is often a trial-and-error process.
2. Escape codes were a new concept, so a learning process concerning their use had to occur. Documentation is sometimes obscure or difficult to acquire for these codes.
3. Interrupts are more complex than is apparent. The code for the interrupt handler was not a problem, but the physical generation of an interrupt signal is complicated. For the IBM PC, for example, the following steps must be taken in order to enable interrupts:

1. The IMR register of the 8259 interrupt controller chip must be set to enable the RS232 interface. This is done via port 21H.
2. The interrupt enable register on the 8250 UART chip must be enabled; this is address 3F9H.
3. OUT2 bit of the modem control register must be set to 1; this is done via port 3FCH.
4. The interrupt vector for the service routine must be installed in 30H.
5. As with all systems, interrupts must be enabled in the program; for the 8088 microprocessor the command is STI.

V. MILITARY REQUIREMENTS AND APPLICATIONS

A. REQUIREMENTS

While the armed forces can and should use civilian networks wherever possible, these must be augmented to meet the special needs driven by the harsh environments in which we operate during wartime. These special needs for survivability, security, precedence, etc., have not been addressed by the ISO or CCITT, so we must develop our own standards and protocols to handle them.

> We are faced with a bothersome fact of military life: the military requirement for communication of data to humans and machines demands that the communications operate at their best precisely when conditions are worst. They must operate when traffic demands far exceed the norms, when enemy action may have destroyed some facilities (which ones cannot be known in advance, of course) and when electronic countermeasures and sabotage are used to attack the system. They must communicate information that could be of great value to an enemy if intercepted and read. Communications management data must not only be protected from intercept (for it can be of considerable intelligence value to an enemy), it must also be protected from "spoofing" by an enemy who wishes to disrupt communications by interjecting false information. Moreover, these concerns apply not merely to tactical systems but to their strategic counterparts as well, since they are not immune from attack either. [Ref. 13:p. 296]

Like the rest of American society, the armed forces are becoming more automated all the time. To compensate for limited numbers of people we are relying more heavily on automation in all areas. Weapons systems of all kinds have microprocessors built into them. The military communications system is based almost completely on computers. All of the headquarters offices in Washington D.C. and elsewhere are computerized for statistical analyses, historical records, personnel assignments, and just about everything else. "Moreover, effective computer communications greatly enhances the effectiveness of automated military operations; we are becoming extremely dependent upon these communications also. There is no turning back." [Ref. 13:p. 296] We automate because it is no longer feasible not to; we cannot perform our jobs adequately any other way. And increasing data communications is the next logical phase of that automation.

B. APPLICATIONS

In many ways, military installations are similar to civilian ones regarding to their needs for data communications. Many offices ashore operate on a sea of paper, creating, filing, forwarding, sometimes even reading tons of paper. The uses of a network to reduce routing time, and to lessen the use of paper as a medium in accordance with the Paperwork Reduction Act of 1980, are only the beginning. As networks grow in sophistication and functionality their usefulness grows also, and military offices need them just as civilian ones do. Memos going throughout a command, or a department, are usually sent with a routing slip from one person to another; this process can take days, and sometimes the news may be obsolete by the time it reaches all recipients. With a network the memo can be routed to

everyone at once; the receipt is instantaneous. Plans of the Day are currently typed, reproduced, and posted and/or routed; again, as with memos, they could be typed into the computer, sent out via the network, then read or printed locally, as desired. Receipt is much faster. At budget time, spreadsheet programs are great aids in preparing budget presentations; these can be sent up the chain via the network, to be incorporated electronically into the command's budget. Ad hoc programs to handle small tasks are often written by people who are not in programming billets. These programs are nevertheless useful, often in many divisions, and sharing them among microcomputers of different vendors without a network is very cumbersome. With a network it becomes a trivial matter, and professional life is made a little easier for everyone.

The U.S. Government has adopted a leading role in instigating and funding research and development in the data communications field, and the Department of Defense is one of the most active players. "The DoD has always had an urgent requirement for data communications and, as a consequence, has spent a considerable amount of resources over the years to determine its precise needs in this area as well as to develop data equipments and systems to meet these needs." [Ref. 12:p. 319] The DoD has many agencies conducting research in data communications, in addition to the ad hoc experimentation that goes on in offices everywhere. These agencies are trying to develop systems not only for strictly military applications, but also systems for applications we have in common with the civilian sector. Some examples:

"Linking a micro to any other micro or mainframe would be the 'piece de resistance' of micro networking. The technology to accomplish it has not fully crystallized yet. But Blackmarr (of Lifson, Herrman, Blackmarr & Harris, dp/OA

consultants based in Dallas) reports that the Pentagon is conducting important experiments with high-capacity broadband LANs that could make this kind of network more readily available for business applications." [Ref. 7:p. 102]

The Department of the Navy Office Automation and Communications System (DONOACS) project team, headed by CDR Robert Gray, is working to develop a Document Interchange Format (DIF). This software will facilitate sharing text files among the office automation terminals of different vendors which are currently incompatible.

> The DIF standard is a 'software filter' implemented by each vendor," said Gray. "In everyday operation, terminal users will rely on normal vendor codes for centering, setting tabs, etc. However, when individuals need to exchange documents with people using other vendors' terminals, they will load vendor-provided DIF software. This software translates the sending terminal's internal code into DIF coding. If the unit relies on codes DIF does not recognize, they simply drop out in communication. On the receiving end, the DIF software transforms the DIF codes into character sets and machine codes native to the terminal. DIF is an intermediate language which exists only in translation. [Ref. 15:p. 49]

The Defense Data Network (DDN) is one of the most ambitious projects ever undertaken in the field of data communications. It was begun as the Arpanet in 1969 by the Defense Advanced Research Projects Agency (DARPA), with the goal of developing "...a flexible and efficient method of communications between computer centers and data terminals." [Ref. 12:p. 319] Bolt Beranek and Newman, Inc was selected as the prime contractor, and still occupies this position. The network started as four nodes, located at the Stanford Research Institute, the University of Utah, and the

University of California campuses at Los Angeles and Santa Barbara. The development was a cooperative effort by universities, private contractors, and government agencies to connect the computers of different manufacturers into a time-sharing system based on packet-switching technology. The software architecture does not correspond directly to ISO's seven layers; there are some variations. The Arpanet project was very successful. It has expanded to include several hundred nodes from Hawaii to Norway, with more planned for the Far East. It has been combined with other networks (MILNET (Military Network) and MINET (Movement Information Network in Europe)) and renamed the Defense Data Network. "Much of our present knowledge about networking is a direct result of the ARPANET project." [Ref. 3:p. 22]

The DDN includes nodes at major American universities and military installations, and has added nodes which are used only for electronic mail. It is a tremendous resource for information sharing. In 1975 it was brought under the management of the Defense Communications Agency (DCA), and it has passed from being experimental to being an operational network. In OPNAV instruction 2070.4 dated 7 March 1984 the Chief of Naval Operations set forth Navy policy regarding use of the DDN; paragraph 2 states that "...The DDN will be used by all DOD ADP systems and data networks requiring interconnection by telecommunications." The Action paragraph states, in part:

> a. This instruction applies to all Navy ADP systems and data networks requiring data communications services. Long-haul and area communications, interconnectivity, and the capability for interoperability will be provided by the DDN. This includes existing ADP systems, ADP systems being expanded and upgraded and new ADP systems.

b. All commands will ensure future ADP acquisitions which require data communications include provisions for using the DDN as their primary data communications medium.

In summation, the armed forces have a long-term need for data communications, and this need has been recognized and addressed at the highest levels of military leadership. The military can make extensive use of civilian facilities, but we must take the lead in fostering the research and development needed for our additional requirements. There are several projects which have already made great contributions to this field, and others going on now which will continue that DOD tradition.

VI. CONCLUSIONS AND RECOMMENDATIONS

"The key is money, training, top-level commitment, and the ability to live with permanent surprise." [Ref. 5:p. W/20]

Permanent surprise is the only unchanging thing in the field of computers. Technology progresses so fast that it is very difficult to keep up. As data communications managers we have to try to establish baseline systems which can be expanded to meet expanding needs, and keep an open mind and a flexible attitude toward new ways of doing things, in order to provide the best systems for our users. In a study of the Hewlett-Packard Company's internal ADP system, one of their executives said "The most significant lesson we have learned from our experience, however, is that there is no one best way to process data. Information systems must be designed to match the organization they support. " [Ref. 16:p. 90]

Just as it is unreasonable to use the same system for all organizations, so is it unreasonable to necessarily limit all applications within an organization to the same system. Some might require the DDN, and others might best be handled by a smaller network like the Lattice Net.

The Lattice Net is a worthwhile project. It is a good vehicle for learning about the inner workings of microcomputers and networks. They are more complicated than they seemed. Each microprocessor and each monitor has its own idiosyncrasies, which must be accommodated in the code. Standards should be used more as guides than as rigid templates.

Recommendations for expansion of the Lattice Net:

1. Add the capability to handle multiple addressees, and mailing lists.

2. Add an on-line message-exchanging capability, so that users can "talk" via the net.
3. Make the interrupt handling routine relocatable, that is, create a mechanism that will allow the routine to locate itself in memory wherever it is most convenient for the system load at the time, so there is no danger of it being overwritten by another program.
4. Create a mechanism to handle the receipt of files by unattended nodes.

The Lattice Net can be a very convenient system to implement and use. The software is free, the transmission medium is already in place, and the network is simple to use. Its functionality will probably remain modest, but it can handle many local needs, and can be very useful.

APPENDIX A

Data Communications Glossary

ASCII - American National Standard Code for Information Interchange. The most widely-used character code for microcomputers for data processing and communications.

asynchronous transmission - a scheme in which data characters are sent at random time intervals. Limits phone-line transmission to about 2400 bps.

baud - signal changes per second. Each change can indicate one, two, or three bits. 1200 bps modems usually operate at 600 baud.

bus network - a system in which all stations, or computer devices, communicate by using passive access to a common distribution channel, or bus.

carrier frequency - a constant signal transmitted between communicating devices that is modulated to encode binary information.

clocking - a technique used to synchronize a sending and a receiving data communications device. Permits synchronous transmission at high speeds.

coaxial cable - a transmission medium, usually employed in local networks. The same medium used by commercial cable television systems.

collision detection - a task performed in a multiple-access network to prevent two computers transmitting at the same time.

database - a repository of information, usually requiring a large computer system with extensive storage capability,

accessed by local and remote terminals for information retrieval. Also called a data bank.

data communications - the entire process and science of enabling digital devices such as computers to communicate with one another.

data packet - a means of transmitting serial data in an efficient package that includes an error-checking sequence.

data rate or data-transfer rate - the speed at which data is sent to a receiving computer - given in bits per second (bps).

dedicated line or leased line - a special telephone line arrangement supplied by the telephone company, and required by certain computers or terminals, whereby the connection is always established.

dial-up line - the normal switched telephone line that can be used as a transmission medium for data communications.

digital - of or relating to the technology of computers and data communications wherein all information if encoded as bits of 1s or 0s that represent on or off states.

direct-connect modem - a device that converts digital signals from a computer into electronic impulses for transmission over telephone lines.

electronic mail or E-mail - a communications service for computer users wherein textual messages are sent to a central computer, or electronic "mailbox", and later retrieved by the addressee.

error-checking or error detection - software routines that identify, and often correct, erroneous data.

full-duplex mode - allows two computers to transmit and receive data at the same time.

half-duplex mode - allows transmission in only one direction at a time; if one device is sending, the other must simply receive data until it's time for it to transmit.

interface - the point of meeting between a computer and an external entity, whether an operator, a peripheral device, or a communications medium. An interface may be physical, involving a connector, or logical, involving software.

local network - one of several short-distance data communications schemes typified by common use of a transmission medium by many devices and high data speeds. Also call a local area network, or LAN.

parallel transmission - eight wires are used to send eight bits (one byte) all at once.

protocol - the rules under which computers exchange information, including the organization of the units of data to be transferred.

ring network - a system in which all stations are linked to form a continuous loop, or circle.

serial transmission - the sending of sequentially ordered data bits.

star network - a system in which all stations radiate from a common controller.

timesharing - a technique that allows more than one terminal to access a central computer simultaneously.

token - a group of bits, such a eight 1s, used in some networks to signal network access by a particular station.

transparency - functions are performed without intervention by the user.

[Ref. 19:pp.118-119]

APPENDIX A

LATTICE NET PROGRAM

```
      .Z80
      ASEG
      ORG 0100H
;*****************************************************
;CONSTANTS:

    BDOS    EQU 0005H
    ESC     EQU 1BH        ;ESCAPE CHAR
    EQL     EQU 3DH        ;= SIGN
    OFFSET  EQU 20H        ;FOR ROW/COL ADDRESSING
    CONOUT  EQU 02H        ;CODE TO PRINT ONE CHAR
    CONIN   EQU 01H        ;CODE TO GET ONE CHAR
    STROUT  EQU 09H        ;CODE TO PRINT STRING
    STRIN   EQU 0AH        ;CODE TO READ STRING
    FCB     EQU 5CH        ;ADDRESS OF FILE CONTROL BLOCK
    ZERO    EQU 00H
    BLANK   EQU 20H
    STAR    EQU 2AH        ;ASTERISK
;*****************************************************
MAIN:
         CALL INIT         ;INITIALIZE FCB
         CALL MENU
         JP   0000H        ;RETURN TO CP/M
;****************************************
;* INITIALIZE THE FILE CONTROL BLOCK     *
;****************************************
INIT:
         PUSH AF
         PUSH BC
         PUSH HL
```

```
        LD   B,35        ;FCB IS 35 BYTES
        LD   HL,FCB
LOOP1:  LD   (HL),ZERO
        INC  HL
        DEC  B
        LD   A,B
        CP   0           ;DONE?
        JP   NZ,LOOP1    ;NO, DO IT AGAIN
        LD   B,11        ;FILENAME IS 11 BYTES
        LD   HL,FCB+1
LOOP2:  LD   (HL),BLANK
        INC  HL
        DEC  B
        LD   A,B
        CP   0           ;DONE?
        JP   NZ,LOOP2    ;NO, DO IT AGAIN
        POP  HL          ;YES, RESTORE REGISTERS
        POP  BC
        POP  AF
        RET
;**************************************************
MENU:
        CALL CLRSCRN
        LD   DE,HDRMSG
        CALL PRSTR       ;PRINT MENU
        CALL GETCHAR     ;GET USER'S MENU SELECTION
        CP   "1"         ;SELECT TO SEND FILE?
        CALL Z,SENDF     ;YES, GOTO ROUTINE TO SEND FILE
        CP   "2"         ;SELECT TO ENTER RECEIVE MODE?
        CALL Z,RECEIVE   ;YES, GOTO RECEIVE ROUTINE
        CP   "3"         ;SELECT TO QUIT?
        JP   Z,DONE      ;YES, JUMP TO DONE
        JP   MENU
DONE:   RET
```

```
;**************************************************
CLRSCRN: PUSH PSW
         LD   E,1AH
         LD   C,02H
         CALL BDOS
         POP  PSW
         RET
;***************************************************
;* PRINT A STRING.                                 *
;* THE ADDRESS OF THE STRING TO PRINT MUST BE      *
;* IN THE DE REGISTER PAIR.                        *
;***************************************************
PRSTR:
         PUSH BC
         PUSH DE
         PUSH HL
         LD   C,09H      ;LOAD CODE TO PRINT STRING
         CALL BDOS
         POP  HL
         POP  DE
         POP  BC

         RET
;*******************************************
;* PRINT THE CHARACTER IN THE A REGISTER. *
;*******************************************
PRCHAR:
         PUSH BC
         PUSH DE
         PUSH HL
         PUSH AF
         LD   C,CONOUT  ;CONOUT=02H
         LD   E,A       ;CHAR TO PRINT
         CALL BDOS
         POP  AF
```

```
        POP  HL
        POP  DE
        POP  BC

        RET
;*********************************************
;* RETURNS ONE CHARACTER IN THE A REGISTER.    *
;*********************************************
GETCHAR:
        PUSH BC
        PUSH DE
        PUSH HL
        LD   C,CONIN    ;CONIN=01H
        CALL BDOS
        POP  HL
        POP  DE
        POP  BC

        RET
;*********************************************
;* GETS READY TO SEND A FILE.                  *
;*********************************************
SENDF:
        LD   HL,ADDR    ;LOAD THE ADDRESS OF THE DESTINATON
        LD   DE,DADDRQ  ;LOAD THE ADDRESS OF DEST. ADDRESS
        CALL PRSTR      ;PRINT THE REQUEST
        CALL GETCHAR    ;GET THE ANSWER
        LD   (HL),A     ;STORE IN MEMORY

        CALL INIT       ;CLEAR FILE CONTROL BLOCK
        LD   DE,FILE    ;PRINT REQUEST FOR FILENAME
        CALL PRSTR
        CALL RDSTR      ;READ ANSWER
                        ;STORE NAME IN FCB
        PUSH BC
        PUSH DE
```

```
        PUSH HL
        LD   A,(BUFCNT)
        INC  A
        LD   B,A          ;# OF BYTES IN NAME
        LD   DE,FCB+1     ;ADDRESS OF FILE NAME
        LD   HL,BUFFER    ;ADDRESS OF STRING BUFFER
STNAME: LD   C,(HL)       ;GET ONE BYTE
        DEC  B
        LD   A,B
        CP   0            ;AM I DONE?
        JP   Z,FINIS      ;YES, JUMP TO RETURN
        LD   A,C          ;NO, CONTINUE
        CP   "."          ;CHECK FOR DELIMITER
        JP   Z,TYPE       ;YES, CHANGE FCB ADDRESS
        PUSH HL           ;SAVE BUFFER ADDRESS
        LD   H,D          ;CAN ONLY USE HL FOR STORING
        LD   L,E
        LD   (HL),C       ;STORE CHAR INTO FCB
        POP  HL           ;RESTORE BUFFER ADDRESS
        INC  HL
        INC  DE
        JP   STNAME
TYPE:   LD   DE,FCB+9     ;ADDRESS OF FILE TYPE
        INC  HL
        JP   STNAME

FINIS:  POP  HL
        POP  DE
        POP  BC
        RET
;*********************************************
RDSTR:
        PUSH BC
        PUSH DE
        PUSH HL
```

```
        LD   DE,BUFMAX
        LD   C,STRIN   ;READ STRING IN
        CALL BDOS
        LD   HL,BUFMAX+2 ;SAVE ADDRESS OF BUFFER
        LD   (BUFMAX-2),HL
        POP  HL
        POP  DE
        POP  BC
        RET
;**********************************************
;* RETURN A CHARACTER FROM STRING BUFFER INTO A.*
;* END-OF-TEXT CHARACTER INDICATES END OF STRING*
;**********************************************
GETSTR:
        LD   A,(BUFCNT) ;LOAD NUMBER OF CHARS IN STRING
        DEC  A          ;DECREMENT
        JP   NC,CONT    ;IF NOT ALREADY DONE, CONTINUE
        LD   A,04H      ;ELSE LOAD EOT TO A
        JP   FINI       ;AND RETURN
CONT:   LD   (BUFCNT),A
        PUSH HL
        LD   HL,(BUFPTR)
        LD   A,(HL)
        INC  HL
        LD   (BUFPTR),HL ;STORE POINTER TO CURRENT CHAR
        POP  HL
FINI:   RET
;********************************************
;* SEND: SENDS CHARACTERS TO THE TELEVIDEO CRT*
;*      WITHOUT USING THE BDOS CALLS.  USED  *
;*      WHEN THE $ USED BY BDOS CAUSES       *
;*      CONFUSION. EXPECTS THE CHAR TO       *
;*      PRINT IN THE C REGISTER.             *
;********************************************
```

```
SEND:
        PUSH BC
        PUSH DE
        PUSH HL
        PUSH AF
CHECK:  IN   A,(3H)      ;INPUT SIO STATUS REGISTER
        BIT  2,A         ;IS DEVICE READY TO RECEIVE?
        JR   Z,CHECK     ;IF NOT, READ AGAIN
        LD   A,C
        OUT  (2H),A      ;ELSE OUTPUT CHARACTER TO CRT
        POP  AF
        POP  HL
        POP  DE
        POP  BC

        RET
;**********************************************
;* RECEIVE: SET INTERRUPT MODE, GET TERMINAL  *
;*          ADDRESS FROM USER, GO TO SLEEP    *
;*          UNTIL INTERRUPTED BY INPUT PORT.  *
;**********************************************
RECEIVE:
        PUSH HL          ;SAVE REGISTERS
        PUSH DE
        IM   1           ;SET INT MODE 1
        DI               ;DISABLE INTERRUPTS
        LD   HL,38H      ;ADDRESS OF INT VECTOR
        LD   (HL),0C3H   ;STORE JUMP INSTRUCTION
        LD   HL,39H
        LD   (HL),00H    ;STORE ADDR OF INT HANDLER
        LD   HL,3AH
        LD   (HL),20H
        LD   DE,ADDRQ    ;REQUEST ADDRESS OF THIS TERMINAL
        CALL PRSTR
        CALL GETCHAR     ;RETRIEVE ONE-BYTE ADDRESS IN A
```

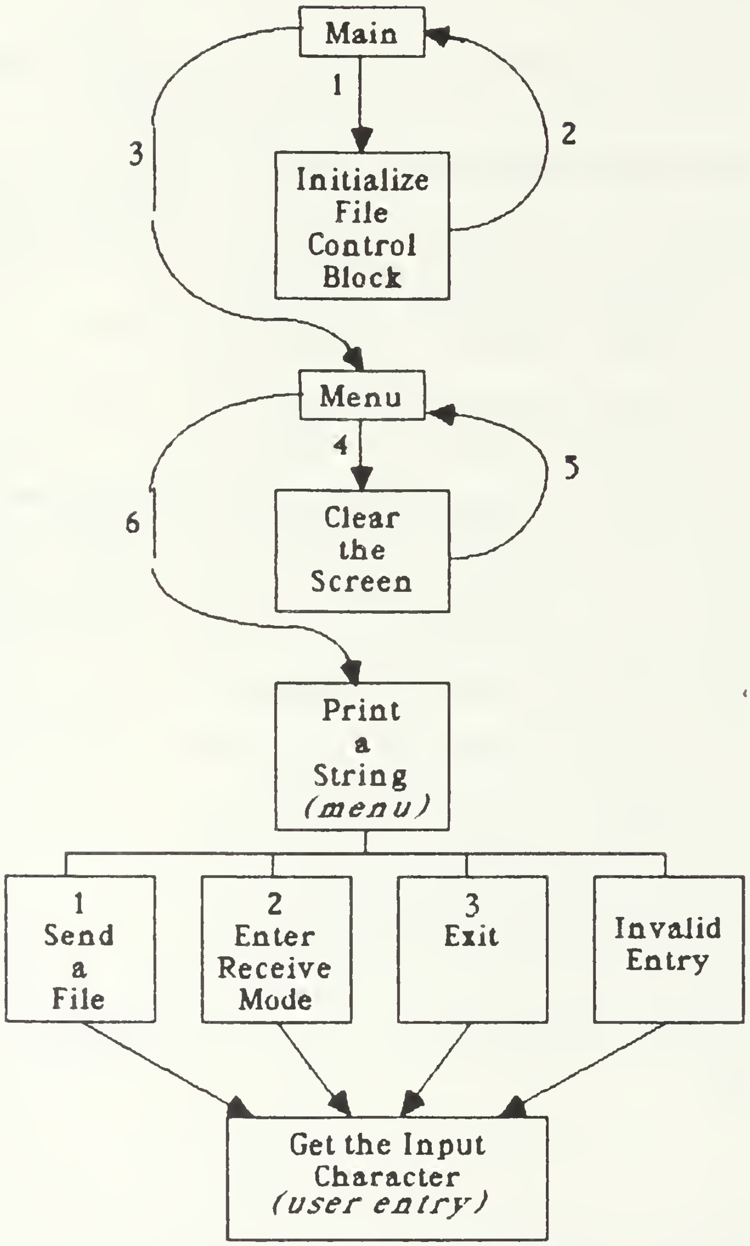
```
        LD   HL,ADDR   ;ADDRESS OF MY ADDRESS
        LD   (HL),A    ;STORE ADDRESS IN MEMORY
        EI             ;ENABLE INTERRUPTS

        POP  DE        ;RESTORE REGISTERS
        POP  HL
        RET
;***********************************************
;* INTERRUPT HANDLER - FOR INCOMING FILES.     *
;***********************************************
        ORG  2000H     ;LOCATE INT HANDLER AT 2000H

        DI             ;DISABLE INTERRUPTS
        PUSH AF        ;SAVE REGISTERS
        PUSH BC
        PUSH DE
        PUSH HL
        PUSH IX
        PUSH IY
        LD   HL,ADDR   ;STORE ADDRESS OF TERMINAL ADDRESS
        LD   B,01      ;HEADER IS 0101
        IN   A,(04H)   ;INPUT BYTE FROM I/O PORT
        CP   B         ;IS IT THE FIRST HEADER BYTE?
        JP   NZ,QUIT   ;IF NO, QUIT
        IN   A,(04H)   ;INPUT NEXT BYTE
        CP   B         ;IS IT THE SECOND HEADER BYTE?
        JP   NZ,QUIT   ;IF NOT, QUIT
        LD   B,(HL)    ;ELSE CHECK FOR MY ADDRESS
        IN   A,(04H)
        CP   B
        JP   NZ,QUIT   ;IF NOT ME, QUIT
                       ;ELSE PRINT "INCOMING" MSG
        LD   HL,INCOMING
QUERY:  LD   A,(HL)
        CP   '       ;CHECK FOR END-OF-STRING SENTINEL
        JR   Z,ANSWER  ;IF END-OF-STRING GET ANSWER
```

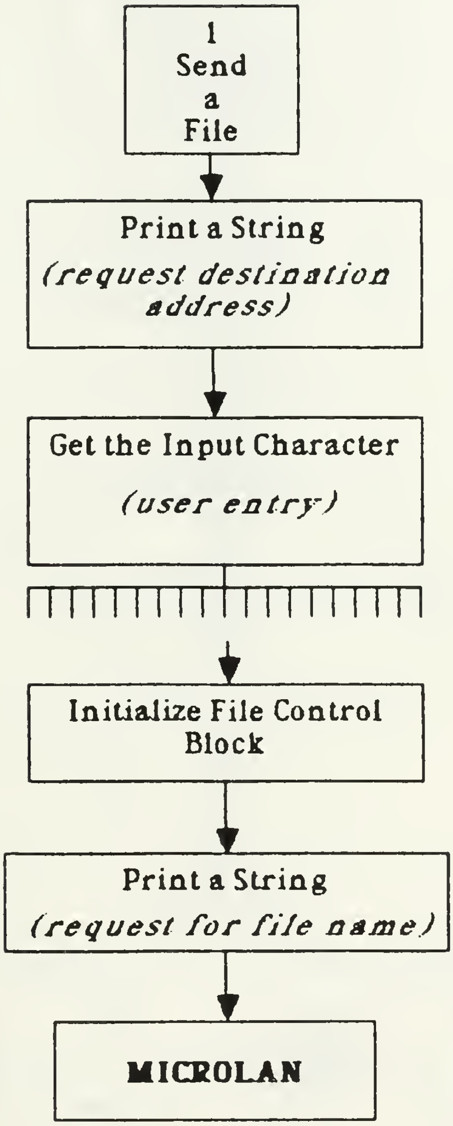
```
        LD   C,A
        CALL SEND       ;SEND CHARACTER TO CRT
        INC  HL
        JR   QUERY
ANSWER: CALL GETCHAR    ;GET RESPONSE
        LD   H,A
        LD   A,"Y"
        CP   H
        JP   NZ,ERASE

ERASE:  LD   HL,CLEAR   ;ERASE "INCOMING" MSG FROM SCREEN
LOP:    LD   A,(HL)
        CP   '        ;CHECK FOR END-OF-STRING SENTINEL
        JR   Z,QUIT     ;IF END-OF-STRING QUIT
        LD   C,A
        CALL SEND       ;SEND CHARACTER TO CRT
        INC  HL
        JR   LOP        ;JUMP TO LOOP

QUIT:   POP  IY
        POP  IX
        POP  HL
        POP  DE
        POP  BC
        POP  AF
        EI              ;ENABLE INTERRUPTS

        RETI            ;RETURN FROM INTERRUPT
;***********************************************
BUFPTR: DW BUFFER
BUFMAX: DB 14
BUFCNT: DS 1
BUFFER: DS 12
HDRMSG: DB ESC,EQL,OFFSET+6,OFFSET+28,'THE LATTICE NET'
        DB ESC,EQL,OFFSET+8,OFFSET+25,'MICROCOMPUTER NETWORK'
        DB ESC,EQL,OFFSET+12,OFFSET+28,'1. SEND A FILE'
```

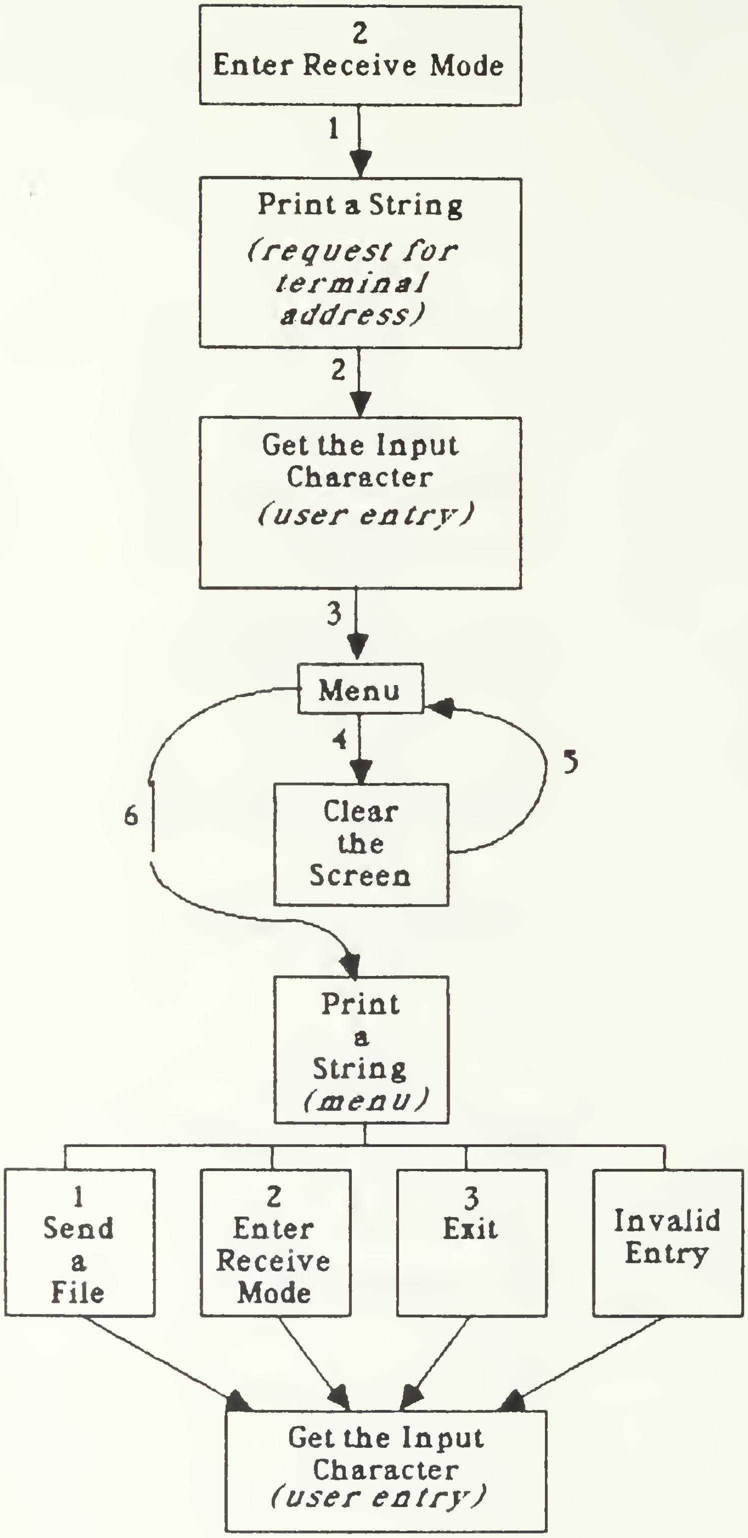
```
        DB ESC,EQL,OFFSET+13,OFFSET+28,'2. ENTER RECEIVE MODE'
        DB ESC,EQL,OFFSET+14,OFFSET+28,'3. EXIT'
        DB ESC,EQL,OFFSET+16,OFFSET+15,'ENTER THE NUMBER OF '
        DB 'YOUR SELECTION:    $'
DADDRQ: DB ESC,EQL,OFFSET+16,OFFSET+15,'ENTER THE DESTINATION
        DB 'ADDRESS (ONE CHARACTER): $'
FILE:   DB 8,8,8,'   '
        DB ESC,EQL,OFFSET+16,OFFSET+15,'ENTER THE NAME OF '
        DB 'THE FILE YOU WANT TO SEND: $'
ADDRQ:  DB ESC,EQL,OFFSET+16,OFFSET+15,'I NEED AN ADDRESS FOR
        DB ESC,EQL,OFFSET+17,OFFSET+15,'PLEASE ENTER ANY SING
        DB ESC,EQL,OFFSET+18,OFFSET+15,'YOU LIKE AS AN ADDRES
INCOMING:DB ESC,EQL,OFFSET+0,OFFSET+49,'*
        DB ESC,EQL,OFFSET+1,OFFSET+49,'*
        DB ESC,EQL,OFFSET+4,OFFSET+49,'*
        DB ESC,EQL,OFFSET+5,OFFSET+49,'*
        DB ESC,EQL,OFFSET+6,OFFSET+49,'**********************
        DB ESC,EQL,OFFSET+2,OFFSET+49,'* YOU HAVE AN INCOMING
        DB ESC,EQL,OFFSET+3,OFFSET+49,'* WILL YOU ACCEPT IT?
CLEAR:  DB ESC,EQL,OFFSET+0,OFFSET+49,'   '
        DB ESC,EQL,OFFSET+1,OFFSET+49,'   '
        DB ESC,EQL,OFFSET+2,OFFSET+49,'
        DB ESC,EQL,OFFSET+3,OFFSET+49,'
        DB ESC,EQL,OFFSET+4,OFFSET+49,'   '
        DB ESC,EQL,OFFSET+5,OFFSET+49,'   '
        DB ESC,EQL,OFFSET+6,OFFSET+49,'
ADDR:   DS 1
;*********************************************
        END
```

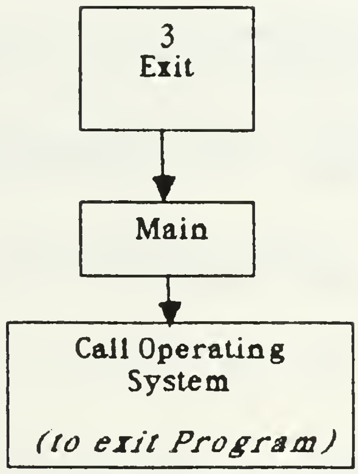
APPENDIX C

Program Hierarchy Chart




1
Send
a
File
Print a String
(request destination address)
Get the Input Character
(user entry)
Initialize File Control Block
Print a String
(request for file name)
MICROLAN


2
Enter Receive Mode
1
Print a String
(request for terminal address)
2
Get the Input Character
(user entry)
3
Menu
4
5
6
Clear the Screen
Print a String
(menu)
1
Send a File
2
Enter Receive Mode
3
Exit
Invalid Entry
Get the Input Character
(user entry)


3
Exit
Main
Call Operating
System
(to exit Program)

LIST OF REFERENCES

1. Staples, Betsy, "Michael Crichton", Creative Computing, February 1985.

2. Sandler, Corey, "Telecommunications Talk", Creative Computing, February 1985.

3. Tanenbaum, Andrew S., Computer Networks, Prentice-Hall, 1981.

4. Blackman, R. Craig, and Cross, Thomas B., "Intelligent Buildings Come of Age", Telecomunication Products + Technology, December 1984.

5. Gantz, John, "Telecommunications: Technology at a Watershed", Telecommunication Products + Technology, October 1984.

6. Scott, Geoff, "The Problem with Standards", Link-Up, August 1984.

7. Foster, Susan, "Networking Micros...Sharing Resources", Computer Decisions, April 1983.

8. Miticelli, Dennis M., and Wright, Michael E., "A Carrier Current Transceiver IC for Data Transmission Over the AC Power Lines", IEEE Journal of Solid State Circuits, Vol. SC-17, No 6, December 1982.

9. Digital Equipment Corporation, Introduction to Local Area Networks, 1982.

10. Jordan, Larry E. and Churchill, Bruce, Communications and Networking for the IBM PC, Robert J. Brady Company, 1983.

11. Theloosen, Henry E., "Integrating PBXs and Computers: An Emerging Reality", Telecommunication Products + Technology, July 1984.

12. Selvaggi, Philip S., "The Department of Defense Data Protocol Standardization Program", Computer Networks, October 1983.

13. Cerf, Vinton G., and Lyons, Robert E., "Military Requirements for Packet-Switched Networks and Their Implications for Protocol Standardization", Computer Networks, October 1983.

14. Latamore, G. Berton, "The Business Connection", _Popular Computing_, March 1985.

15. Government Business News, January 1985.

16. Van Rensselaer, Cort, "Centralize? Decentralize? Distribute?", _Datamation_, April 1979.

17. King, John Leslie, "Centralized versus Decentralized Computing: Organizational Considerations and Management Options", _Computing Surveys_, December 1983.

18. Cash, James I. Jr., Mcfarlan, F. Warren, and McKenney, James L., _Corporate Information Systems Management: Text and Cases_, Richard D. Irwin, Inc., 1983.

19. Byers, T.J., "Micro-To-Micro Communications", _Popular Computing_, February 1984.

BIBLIOGRAPHY

Apple Computer Inc., Apple Reference Manual, 1982.

"Choosing and Using Telecommunications Software", Creative Computing, February 1985, pp. 100-113.

Hogan, Thom, Osborne CP/M User Guide, Osborne/McGraw-Hill, 1982.

Miller, Alan R., Mastering CP/M, Sybex, Inc, 1983.

Zaks, Rodnay, Programming the Z80, Sybex Inc., 1982.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Commander, Naval Data Automation Command Washington Navy Yard Washington, DC 20374	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, CA 93943	2
3.	Superintendent, Code 53Lz Attn: G. Latta Naval Postgraduate School Monterey, CA 93943	1
5.	Superintendent, Code 54Lb Attn: N.R. Lyons Naval Postgraduate School Monterey, CA 93943	1
6.	Superintendent, Code 37 Computer Technology Programs Curricular Officer Naval Postgraduate School Monterey, CA 93943	1
7.	LT Alex Calder VQ3 NAS Barbers Point, Hawaii 96862	1
8.	Superintendent, Code 0141 Attn: A. Austin Naval Postgraduate School Monterey, CA 93943	1
9.	Naval Data Automation Command Attn: LCDR J.L. Egbert Washington Navy Yard Washington, DC 20374	3
10.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2

211219

Thesis
E2642 Egbert
c.1 Microcomputer networks: investigation and implementation.